

Powershell

Astuces

- Exécution des scripts:

```
set-executionpolicy remotesigned
```

- [Encodage UTF-8](#), sinon convertir en UTF-8 avec BOM
- [Sorties sous forme de tableaux non tronqués](#):

```
Get-WuaHistory | Sort-Object Date | Format-Table -Property * -AutoSize  
| Out-String -Width 4096
```

- Avec Windows 7 nous pouvons être bloqué à la version 2 de Powershell

```
Get-Host | Select-Object Version
```

même si le dernier framework .NET 4.* est installé, du coup impossible d'accéder à certaines fonctions comme

```
[string]::IsNullOrEmpty($SomeString)
```

2 solutions:

- réécrire la fonction:

```
# PowerShell v2.0 compatible version of  
[string]::IsNullOrEmpty.  
function StringIsNullOrEmpty([string] $string)  
{  
    if ($string -ne $null) { $string = $string.Trim() }  
    return [string]::IsNullOrEmpty($string)  
}
```

- [Mettre à jour Powershell](#)
- [Browse the .NET Framework source code online, with search and navigation powered by Roslyn.](#), peut servir

Scripts

Liste MAJ Windows

Objectif

Singer la GUI de WUpdate (date, status, numéro, description, url) avec un tri par date et limiter la sortie, export en texte

Ressources

- <https://stackoverflow.com/questions/41626129/how-do-i-get-the-update-history-from-windows-update-in-powershell>
- <https://blogs.technet.microsoft.com/tmintner/2006/07/07/powershell-to-see-what-updates-have-been-installed/>
- <http://tomtalks.uk/2013/09/list-all-microsoftwindows-updates-with-powershell-sorted-by-kbhotfix-id-get-microsoftupdate/>

Code

```
# conversion codes wua
function Convert-WuaResultCodeToName {
    param(
        [Parameter(Mandatory=$true)]
        [int] $ResultCode
    )
    $Result = $ResultCode
    switch($ResultCode) {
        0 {
            $Result = "Non débutée"
        }
        1 {
            $Result = "En cours"
        }
        2 {
            $Result = "Réussite"
        }
        3 {
            $Result = "Réussite avec des erreurs"
        }
        4 {
            $Result = "Echec"
        }
        5 {
            $Result = "Avortée"
        }
    }
    return $Result
}

# listing
function lister-MAJ {
    # test paramètre
    param(
        [Parameter(Mandatory=$false)]
        [int] $quantite
    )
}
```

```

)

$wu = new-object -com "Microsoft.Update.Searcher"

If ($quantite -eq 0) {
    # si absence de paramètre, l'enemble des MAJ
    $totalupdates = $wu.GetTotalHistoryCount()
} else {
    $totalupdates = $quantite
}

$all = $wu.QueryHistory(0,$totalupdates)

# Define a new array to gather output
$OutputCollection= @()
Foreach ($update in $all) {
    $date = $update.date
    $title = $update.title
    $state = Convert-WuaResultCodeToName -ResultCode $update.resultcode

    # KB + n'importe quel nombre de digit
    $Regex = "KB\d+"
    $KB = $title | Select-String -Pattern $regex | Select-Object {
$_.Matches }
    $output = New-Object -TypeName PSObject
    $output | add-member NoteProperty "Date" -value $date
    $output | add-member NoteProperty "Title" -value $title
    $output | add-member NoteProperty "Etat" -value $state
    $output | add-member NoteProperty "HotFixID" -value $KB.'
$_.Matches '.Value

    $OutputCollection += $output
}
return $OutputCollection | Sort-Object Date | Format-Table -AutoSize |
Out-String -Width 4096
}
# prend en compte 1er argument passé au script
lister-MAJ -quantite $args[0]

```

Exemple de sortie:

```
p> lister_maj.ps1
```

```

Date                Title
Etat                HotFixID
----                -
-----
30/12/1899 00:00:00
Non débutée
30/12/1899 00:00:00
Non débutée

```

```

30/12/1899 00:00:00
Non débutée
20/03/2019 13:05:51 2018-04 Mise à jour cumulative pour Windows 10 Version
1607 pour les systèmes x64 (KB4093120) Avortée KB4093120
20/03/2019 14:00:30 Mise à jour pour Windows 10 Version 1607 pour
ordinateurs à processeur x64 (KB4049411) Avortée KB4049411
20/03/2019 14:09:56 Update for Windows 10 Version 1607 for x64-based Systems
(KB4480730) Avortée KB4480730
20/03/2019 15:24:52 Mise à jour pour Windows 10 pour ordinateurs à
processeur x64 (KB4056254) Réussite KB4056254
20/03/2019 15:25:21 Mise à jour pour Windows 10 Version 1607 pour
ordinateurs à processeur x64 (KB4049411) Réussite KB4049411
20/03/2019 15:25:44 Mise à jour pour Windows 10 Version 1607 pour
ordinateurs à processeur x64 (KB4033631) Réussite KB4033631
20/03/2019 15:26:07 2018-05 Mise à jour pour Windows 10 Version 1607 sur
systèmes x64 (KB4132216) Réussite KB4132216
20/03/2019 15:30:01 Outil de suppression de logiciels malveillants Windows
x64 - mars 2019 (KB890830) Réussite KB890830
20/03/2019 15:30:15 Update for Windows 10 Version 1607 for x64-based Systems
(KB4480730) Réussite KB4480730
20/03/2019 15:30:25 2019-03 Mise à jour pour Windows 10 Version 1607 sur
systèmes x64 (KB4023057) Echec KB4023057
20/03/2019 17:04:09 2018-04 Mise à jour cumulative pour Windows 10 Version
1607 pour les systèmes x64 (KB4093120) Avortée KB4093120
20/03/2019 17:31:54 Mise à jour de sécurité pour Windows (KB4093119)
Réussite KB4093119
20/03/2019 20:58:38 2018-04 Mise à jour cumulative pour Windows 10 Version
1607 pour les systèmes x64 (KB4093120) Réussite KB4093120
20/03/2019 20:59:00 2018-06 Mise à jour pour Windows 10 Version 1607 sur
systèmes x64 (KB4134659) Réussite KB4134659
20/03/2019 20:59:08 2019-03 Mise à jour pour Windows 10 Version 1607 sur
systèmes x64 (KB4023057) Echec KB4023057

```

Supprimer apps

Objectif

Supprimer les applications (apps) de Windows 10 de l'image de référence (ce qui est forgé avec MDT), ce qui permettra d'éviter qu'elle se réinstallent à chaque MAJ système et/ou à chaque création d'utilisateur.

Ressources

- Basé sur <https://deploymentparts.wordpress.com/2015/09/10/remove-builtin-apps-from-windows-10-reference-image/>
- Lui-même inspiré de

<https://github.com/W4RH4WK/Debloat-Windows-10/blob/master/scripts/remove-default-apps.ps1>

- Doc de Remove-AppxPackage
<https://docs.microsoft.com/en-us/powershell/module/appx/remove-appxpackage?view=win10-ps>
- Une autre doc <https://www.askvg.com/guide-how-to-remove-all-built-in-apps-in-windows-10/>

Code

```
# lister les app de l'utilisateur  
courant, trié par nom
```

```
# lister les app de tous les  
utilisateurs, trié par nom
```

```
# lister les app de l'image de  
référence
```

```
# le script qui va bien
```

From:

<https://wiki.xanatos.net/> - **Base de connaissances**

Permanent link:

<https://wiki.xanatos.net/doku.php?id=windows:powershell>

Last update: **2019/03/21 09:11**

